

图文 010、动手实验：亲自感受一下线上系统部署时如何设置JVM内存

4156 人次阅读 2019-07-10 07:00:00

详情 评论

---

### 动手实验：亲自感受一下线上系统部署时如何设置JVM内存大小？

---

给大家推荐一套质量极高的Java面试训练营课程：



作者是中华石杉，石杉老哥是我之前所在团队的 Leader，骨灰级的技术神牛！

大家可以点击下方链接，了解更多详情，并进行试听：

[21天互联网Java进阶面试训练营（分布式篇）](#)

---

#### 重要说明：

最近不少同学留言反馈，说希望建立一个微信群，供大家进行JVM专栏的学习交流。

这个提议非常好，不过管理微信群是一件挺费时的事儿，我平时工作较忙，实在抽不出时间来进行群管理。

正好石杉老哥的面试训练营建了微信交流群，并且还请了不少一线大厂的助教。

因此跟石杉老哥商量了一下，决定厚着脸皮“鸠占鹊巢”。购买了我JVM专栏的小伙伴，可以加入石杉老哥的微信群，在群里讨论交流技术。

**如何加群，请参见文末（注：如果之前已经加过的，就不要重复加群了）**

---

推荐结束，正文开始

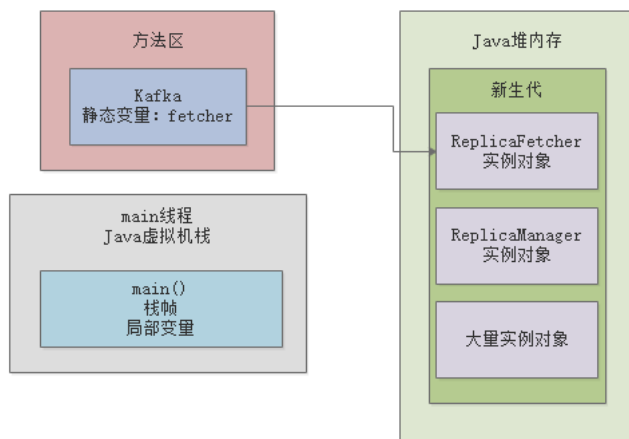
## 目录:

- 前文回顾
- 跟JVM内存相关的几个核心参数图解
- 如何在启动系统的时候设置JVM参数?
- 通过案例, 学习参数优化设置的预告
- 昨日思考题解析
- 今日思考题

## 1、前文回顾

咱们先简单回顾一下目前为止已经学到的内容, 现在大家肯定都知道, 我们平时代码里创建的对象, 都是优先在新生代分配的

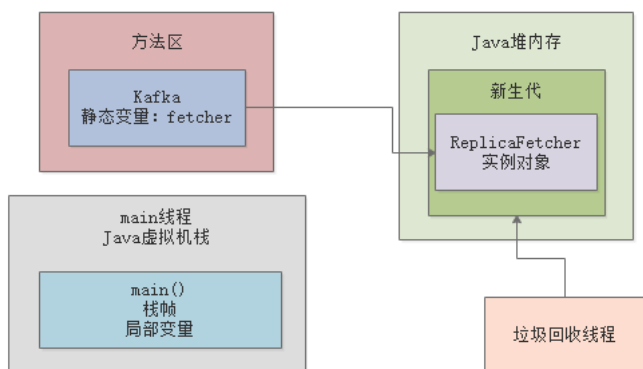
然后随着一些方法执行完毕, 大部分新生代里的对象就没有人引用了, 就成了垃圾对象, 如下图所示:



大家可以想象一下, 随着代码持续运行, 新生代里对象会越来越多, 而且里面大部分的对象其实都是那种短生存周期的对象, 很快就没有人引用他们了, 因此大部分都会是一些垃圾对象。

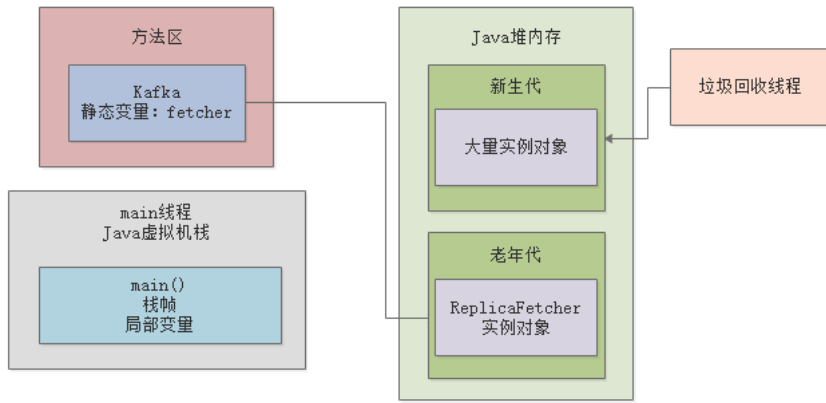
然后代码继续运行, 是不是肯定会创建新的对象分配在新生代里?

肯定是的, 所以一旦新生代里内存不够了, 就会触发一次Minor GC, 把新生代里那些没有人引用的垃圾对象都给回收掉, 腾出来内存空间。如下图所示:



如果是那种在长周期存活的对象, 他在新生代里会持续躲过多次垃圾回收, 每躲过一次垃圾回收, 年龄会增长1岁

然后当他成为是十多岁的“老年人”的时候, 就会被转移到老年代里去, 如下图:



好，到此为止，我们撇开一些特殊情况，至少JVM中基本的内存分配原理，大家就搞清楚了

本周最核心的概念，就是让大家厘清楚短生存周期的对象和长生存周期的对象分别是什么，他们是如何在新生代里分配的，新生代什么时候触发Minor GC，然后长生存周期的对象如何转移到老年代里去。

## 2、跟JVM内存相关的几个核心参数图解

本文就不涉及到任何原理性的东西了，直接开始逐步给大家讲解JVM的参数如何设置。

在JVM内存分配中，有几个参数是比较核心的，如下所示。

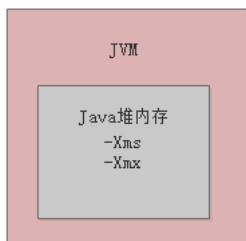
- Xms: Java堆内存的大小
- Xmx: Java堆内存的最大大小
- Xmn: Java堆内存中的新生代大小，扣除新生代剩下的就是老年代的内存大小了
- XX:PermSize: 永久代大小
- XX:MaxPermSize: 永久代最大大小
- Xss: 每个线程的栈内存大小

下面我们对上述参数来进行一一说明。

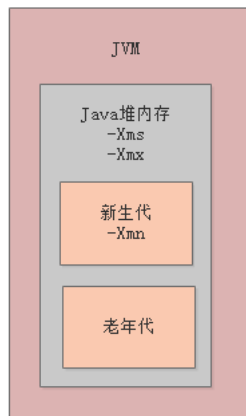
-Xms和-Xmx，分别用于设置Java堆内存的刚开始的大小，以及允许扩张到的最大大小。

对于这对参数，通常来说，都会设置为完全一样的大小。大家先不用太过于纠结这里的细节，因为其实JVM里的各种技术细节真的太多了，不能一下子全部都搞定，要随着后续几十个案例，层层铺展开来。

但是至少大家需要清楚，这两个参数，是用来**限定Java堆内存的总大小的**，如下图。



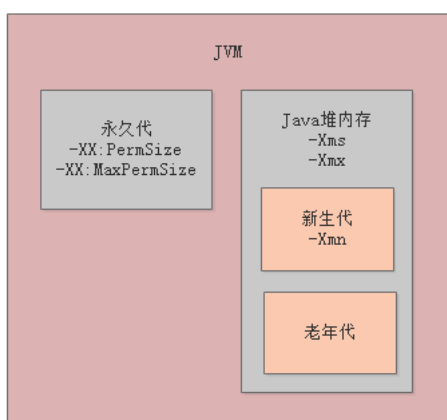
-Xmn，这个参数也是很常见的，他用来**设置Java堆内存中的新生代的大小**，然后扣除新生代大小之后的剩余内存就是给老年代的内存大小，我们看下图：



-XX:PermSize和-XX:MaxPermSize，分别**限定了永久代大小和永久代的最大大小**

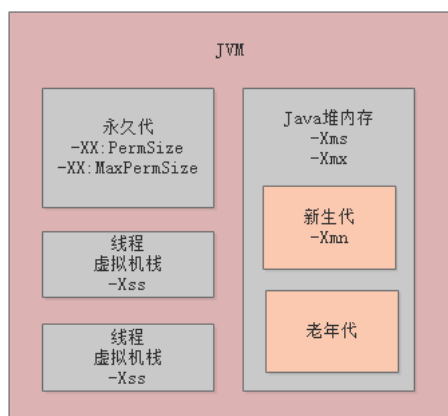
通常这两个数值也是设置为一样的，至于原因，请看后面结合案例的文章分析。

如果是JDK 1.8以后的版本，那么这俩参数被替换为了-XX:MetaspaceSize和-XX:MaxMetaspaceSize，但是大家至少得知道，这两个参数限定了永久代的大小，如下图所示：



-Xss，这个参数**限定了每个线程的栈内存大小**

大家都很清楚，每个线程都有一个自己的虚拟机栈，然后每次执行一个方法，就会将方法的栈帧压入线程的栈里，方法执行完毕，那么栈帧就会从线程的栈里出栈，如下图：



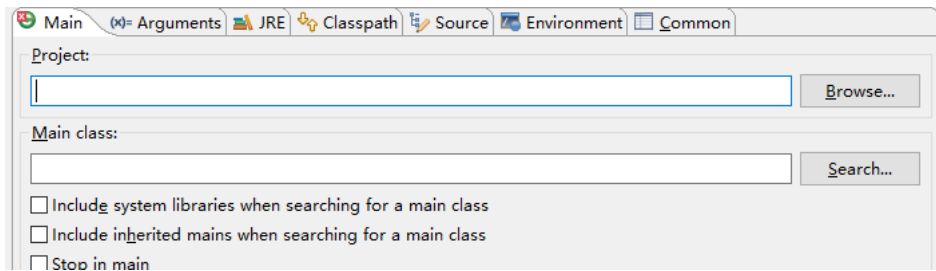
### 3、如何在启动系统的时候设置JVM参数？

那么现在大家结合图示都知道了JVM内存各个区域的大小该使用什么参数来设置，那么到底怎么设置呢？

现在就带着大家来做一些实验

你要是在Eclipse/IntelliJ IDEA里开发代码的话，如果要在这种开发IDE里启动一个程序，然后设置JVM参数，那么就需要对按照下面的步骤来设置：

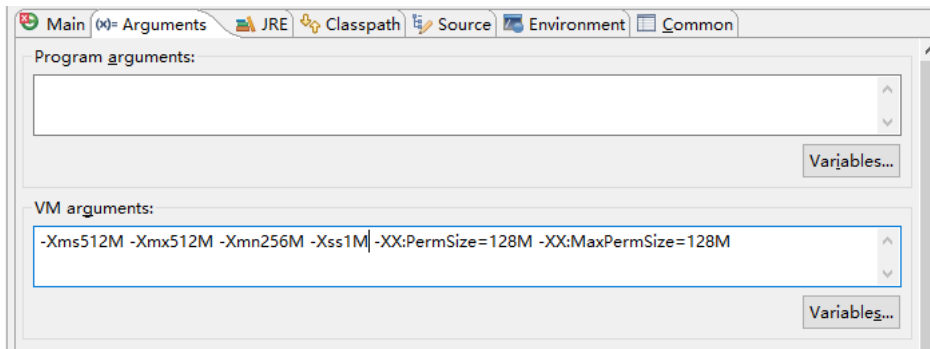
首先右击你写好的一个带main()方法的类，他有一个菜单栏，里面有一个“**Debug as**”选项，鼠标移动进入，会看到一个“**Debug Configuration**”选项，接着会看到下面的面板。



这个面板里有一个“**Arguments**”的选项，点击他，会看到下面的图。然后在“VM arguments”中输入你的JVM参数即可

比如你可以按照下面的示例来设置，-Xms之类的参数直接后面跟上你要设置的内存大小，多少M即可。

但是-XX:PermSize这种格式的参数，需要跟一个“=”符号，跟上你要设置的内存大小即可。



那么如果是在[线上部署系统](#)应该如何设置JVM参数呢？

其实都很简单，比如说采用“java -jar”的方式启动一个jar包里的系统，那么就可以采用类似下面的格式：

```
java -Xms512M -Xmx512M -Xmn256M -Xss1M -XX:PermSize=128M -XX:MaxPermSize=128M -jar App.jar
```

如果是现在非常流行的那种启动Spring Boot开发的系统呢？

其实都是类似的，大家自行翻阅一下Spring Boot的文档即可。

#### 4、通过案例，学习参数优化设置的预告

明天开始，我们会做发布本专栏的第一个案例，就是百万交易的支付系统案例

通过分析一个支付系统的核心业务流程，然后结合我们学习到的JVM相关的知识，来一步步探究，JVM内存相关的这些核心参数，到底在我们上线一个生产系统的时候，针对预估的并发压力，到底应该如何合理的给出一个未经过调优的比较合理的初始值。

另外我们会分析各种参数在设置的时候有哪些考虑的点，Java堆内存到底需要多大？新生代和老年代的内存分别需要多大？永久代和虚拟机栈分别需要多大？这些我们都结合案例来一步一步的分析。

其实JVM参数到底该如何设置，一定是根据不同的业务系统他具体的一些场景来调整的，不是说有一个通用的配置和模板，照着设就没问题了，那个思路肯定是不对的，也不能干巴巴的告诉你，这个参数应该这样设置，那个参数应该那样设置。

一切都要从案例出发，结合业务场景来分析。

## 5、昨日思考题解析

昨天让大家结合学到的知识，去分析一下自己手头负责的那些业务系统，哪些是短生存周期的对象，哪些是长生存周期的对象

其实就是让大家开始在脑子里建立起来自己负责的系统在JVM中运行时的一个概念图，要有这个意识，才能更好的进行JVM调优。

## 6、今日思考题

给大家留一个思考题，大家都知道平时我们一般部署Java系统，主要以Tomcat部署Web系统居多，现在还更多的是通过Spring Boot来部署系统。

那么大家可以自行去网上查阅一下，Tomcat、Spring Boot部署启动系统的时候，JVM参数如何设置？

然后大家再看看自己公司手头负责的系统，你们部署系统的时候是通过什么方式来设置JVM参数的？

**End**

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

---

### 如何加群？

- 1、添加微信号：Giotto1245 （微信名：Jarvis）
- 2、发送 Jvm专栏的购买截图
- 3、人工操作，发送截图后请耐心等待被拉群

**最后提醒：**之前加过面试群的同学就不要重复加了

---

## 常见问题解答：

### 一、如何生成自己的分享海报并获取返现？

#### 方式1：

点击文章右上角**邀请好友**（如下图），生成自己的专属海报。

将海报发送给好友或分享朋友圈，朋友通过扫描你分享的海报购买课程，你将**获取返现24元**，可在个人中心中提现：

**累计邀请30人**，你将升级为高级推广员，此后每成功邀请一位朋友，返现翻倍。换句话说，从第31人开始，每成功邀请一位朋友，你将**获取返现48元**